



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/853,226	05/11/2001	Geoffrey S. Strongin	2000.039300/TT3766	6345
23720	7590	09/29/2005	EXAMINER	
WILLIAMS, MORGAN & AMERSON, P.C. 10333 RICHMOND, SUITE 1100 HOUSTON, TX 77042			RIZZUTO, KEVIN P	
		ART UNIT	PAPER NUMBER	
		2183		

DATE MAILED: 09/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/853,226	STRONGIN, GEOFFREY S.
	Examiner Kevin P. Rizzuto	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 26 July 2005.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-53 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-53 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.

2. Certified copies of the priority documents have been received in Application No. _____.

3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____

5) Notice of Informal Patent Application (PTO-152)

6) Other: _____

DETAILED ACTION

1. Claims 1-53 have been examined.
2. Acknowledgement of papers filed: Amendment filed on 7/26/2005. The papers filed have been placed on record.
3. The 35 U.S.C. 102 Rejections to claims 25-26, 29-30, 36, 43-44, 47-48 for being anticipated by Sakaki (U.S. Patent 5,826,007) and the 35 U.S.C. 103 Rejections to claims 31, 37-40 and 49 for being unpatentable over Sakaki, in view of Kime, have been withdrawn by the examiner. Examiner notes that the wrong reference was unintentionally cited in the original rejection, all references to Sakaki were intended to be directed towards Yishay et al., U.S. Patent 5,704,039. The new 35 U.S.C. 102 and 103 Rejections are found below with the corrected references.

Maintained Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
5. Claims 1-2, 5-17, 21-24, 27-28, 32-34, 45-46 and 50-52 are rejected under 35 U.S.C. 102(b) as being anticipated by Gafken, U.S. Patent 6,026,016.
6. As per claim 1, Gafken teaches a processor, comprising:

-A first register configured to store one or more hardware-debug-test (HDT) enable bits: (Column 8, lines 21-44, figure 7, R/L register, if the R/L register is set to enable a read, it is set to enable a hardware-debug-test, because a read is an example of a hardware-debug-test. In normal debugging of a processor, multiple registers or memory locations are tested by having their contents examined, i.e., a read of their data is performed. Therefore, a read of a memory location is a hardware-debug-test. Furthermore, if the memory read operation is not unlocked, a hardware debug test on the memory is not possible, since it cannot be determined what is actually in the memory without the read operation.

Therefore, enabling a read operation of memory is also enabling a hardware-debug-test. Gafken also states that the locks should be easy to unlock in order to fix bugs in code stored in locked blocks, i.e., unlocking the memory locations is enabling debugging to occur. (Col. 2, lines 7-10.)

-A first control logic coupled to receive a plurality of HDT input signals, wherein the first control logic is coupled to access the first register: (Interface 710 determines whether the R/L bit is set upon a read operation.) (Figure 7, Column 8, lines 30-32)

-And a second control logic coupled to the first register, wherein the second control logic is configured to store one or more default values in the first register in response to a reset of the processor. (Figure 7, Column 11, lines 38-55, the reset detector outputs a signal to assert all the read lock bits (HDT enable bits))

7. As per claim 2, Gafken teaches the processor of claim 1,

-Wherein the first control logic is further configured to receive a request to enter an HDT mode: (The interface unit 710 receives all memory operations, including memory read operations (a request to enter a HDT mode). Column 5, lines 60 to column 6, line 3 and column 8, lines 21-44)

-Wherein the first control logic is further configured to read selected entries of the one or more HDT enable bits stored in the first register in response to the request to enter HDT mode: (Column 7, lines 47-54, and column 8, lines 21-44).

-And wherein the first control logic is further configured to grant or deny the request to enter HDT mode based on the selected entries of the one or more HDT enable bits: (Column 7, lines 47-54, and column 8, lines 21-44).

8. As per claim 5, Gafken teaches the processor of claim 1, wherein the second control logic is further coupled to receive a signal (signals labeled, "From Bus" in figure 7) indicative of the one or more default values for the one or more HDT enable bits and to write the one or more default values for the one or more HDT enable bits into the first register in response to the reset of the processor. (Gafken teaches wherein the R/L bits (HDT enable bits) are loaded with default values after a reset that is detected by the Reset Detector, which receives signals that indicate a reset. Therefore, the Reset Detector receives a signal indicative of the default values for the HDT enable bits. (Column 5, lines 47-59, column 8, lines 21-44 and column 10, line 62 to column 11, lines 3-55)).

9. As per claim 6, Gafken teaches the processor of claim 1, wherein the second control logic is coupled to receive a RESET signal in response to the reset of the processor. (Figure 7, column 5, lines 47-59, column 8, lines 21-44 and column 10, line 62 to column 11, lines 3-55. Reset Detector is coupled to receive a RESET signal)

10. As per claim 7, Gafken teaches the processor of claim 1, further comprising:

-A third register configured to store one or more microcode loader enable bits: (W/L register, figure 7, column 6, lines 20-50, column 8, lines 21-41, column 1, lines 13-31)

-A third control logic coupled to receive a plurality of microcode inputs, wherein the third control logic is coupled to access the third register: (Interface 710 contains both logic to control the access to the R/L register and logic to control the W/L (microcode loader enable) register. Since the interface 710 can access both the R/L and W/L registers individually, and while it is unclear how much of the control logic for the R/L control and for the W/L control is unique to each control logic portion respectively, the interface 710 inherently has at least some control logic for the R/L control that is separate from the control for the W/L control. The shared logic (including inputs) to interface 710 along with the inherent, not shown, unique W/L control logic makes up the third control logic. (Figure 7, Column 8, lines 21-44)

-And a fourth control logic coupled to the third register, wherein the fourth control logic is configured to store one or more default values in the third register in response to a reset of the processor. (Reset Detector contains both logic to store default values to the R/L register and to the W/L (microcode loader enable) register. Since the Result Detector has connections to both the R/L and W/L registers individually (shown in figure 7), and while it is unclear how much of the Result Detector logic is unique for the R/L control and how much is unique for the W/L control, the RESET Detector inherently has at least some control logic for the R/L control that is separate from the control for the W/L control. The shared logic (including inputs) to Reset Detector along with individual physical connections to W/L register shown in figure 7 makes up the fourth control logic. (Figure 7, column 10, line 62 to column 11, line 55)

11. As per claim 8, Gafken teaches the processor of claim 7, wherein the third control logic (Interface 710) is further configured to receive a request to modify microcode, wherein the third control logic is further configured to read selected entries of the one or more microcode loader enable bits stored in the third register (W/L) in response to the request to modify microcode, and wherein the third control logic is further configured to grant or deny the request to modify microcode based on the selected entries of the one or more microcode loader enable bits. (Column 6, lines 20-50, column 7, lines 47-54, column 8, lines 30-44)

12. As per claim 9, Gafken teaches the processor of claim 1, further comprising: a second register (L/D) coupled to the first control logic, wherein the second register is

configured to store one or more HDT enable lock bits. (Column 8, lines 21-44, column 6, lines 51-59; when in lock-down mode, the HDT enable bit (R/L) associated with the lock-down bit, is not modifiable.)

13. As per claim 10, Gafken teaches the processor of claim 9,

-Wherein the first control logic (Interface 710, figure 7) is further configured to receive a request to modify HDT mode status: (Column 8, lines 42-44, column 7, lines 47-54, column 6, lines 19-59 and column 2, line 65 to column 3, line 12)

-Wherein the first control logic is further configured to read selected entries in the one or more HDT enable lock bits stored in the second register in response to the request to modify HDT mode status: (Column 8, lines 42-44, column 7, lines 47-54, column 6, lines 19-59 and column 2, line 65 to column 3, line 12)

-And wherein the first control logic is further configured to grant or deny the request to modify HDT mode based on the selected entries in the one or more HDT enable lock bits. (Column 8, lines 42-44, column 7, lines 47-54, column 6, lines 19-59 and column 2, line 65 to column 3, line 12)

14. As per claim 11, Gafken teaches the processor of claim 9, wherein the first register and the second register are unified into a single register configured to store two or more bits, including one or more HDT enable bits and one or more HDT enable lock bits (Figure 7, the W/L and L/D registers are unified as they are both within the Lock Bit Array 705 and also because they are both within a Segment N containing unified R/L, W/L, and L/D bits.)

15. As per claim 12, The processor of claim 9, further comprising:

-A third register configured to store one or more microcode loader enable bits:

(The W/L register of figure 7 stores one or more microcode loader enable bits.

Column 6, lines 20-50, column 8, lines 21-41, column 1, lines 13-31)

-A third control logic coupled to receive a plurality of microcode inputs, wherein the third control logic is coupled to access the third register: (Interface 710 contains both logic to control the access to the R/L register and logic to control the W/L (microcode loader enable) register. Since the interface 710 can access both the R/L and W/L registers individually, and while it is unclear how much of the control logic for the R/L control and for the W/L control is unique to each control logic portion respectively, the interface 710 inherently has at least some control logic for the R/L control that is separate from the control for the W/L control. The shared logic (including inputs) to interface 710 along with the inherent, not shown, unique W/L control logic makes up the third control logic.

(Figure 7, column 8, lines 21-44)

-And a fourth control logic coupled to the third register, wherein the fourth control logic is configured to store one or more default values in the third register in response to a reset of the processor: (Reset Detector contains both logic to store default values to the R/L register and to the W/L (microcode loader enable) register. While it is unclear how much of the Result Detector logic is unique for the R/L control and how much is unique for the W/L control, the Result Detector

has connections to both the R/L and W/L registers individually (shown in figure 7), and therefore RESET Detector inherently has at least some control logic for the R/L control that is separate from the control for the W/L control. The shared logic (including inputs) to Reset Detector along with individual physical connections to W/L register shown in figure 7 makes up the fourth control logic. (Figure 7, column 10, line 62 to column 11, line 55)

16. As per claim 13, Gafken teaches the processor of claim 12, wherein the third control logic (Interface 710) is further configured to receive a request to modify microcode, wherein the third control logic is further configured to read selected entries in the one or more microcode loader enable bits stored in the third register (W/L) in response to the request to modify microcode, and wherein the third control logic is further configured to grant or deny the request to modify microcode based on the selected entries in the one or more microcode loader enable bits. (Column 6, lines 20-50, column 7, lines 47-54, column 8, lines 30-44)
17. As per claim 14, Gafken teaches the processor of claim 12, wherein the second and fourth control logics are unified. (Figure 7 shows the second and fourth control logic are unified by the Reset Detector block)
18. As per claim 15, Gafken teaches the processor of claim 14, wherein the first control logic, the second control logic, the third control logic, and the fourth control logic are unified. (Figure 7 shows that the first, second, third and fourth control circuit are all located on the flash memory device 700, and are therefore all unified.)

The first, second, third and fourth are also unified because they are connected directly by an unlabeled wire shown in figure 7.)

19. As per claim 16, Gafken teaches a processor, comprising:

-A first control logic (Interface 139) coupled to receive a plurality of microcode inputs ("From Bus"): (Figure 3, figure 5, step 565, column 1, lines 18-31, column 5, line 60 to column 6, line 3, column 8, lines 53 to column 9, line 2, column 13, lines 7 to column 14, lines 25)

-A first register coupled to the first control logic, wherein the first register is configured to store one or more microcode loader enable bits: (Figures 3 and 5, column 6, lines 19-50, the bits in W/L registers are microcode loader enable bits)

-And a second control logic coupled to the first register, wherein the second control logic is configured to store one or more default values in the first register in response to a reset of the processor. (Figure 3, Reset Detector, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55)).

20. As per claim 17, Gafken teaches the processor of claim 16, wherein the first control logic is further configured to receive a request to modify microcode, wherein the first control logic is further configured to read selected entries of the one or more microcode loader enable bits stored in the first register in response to the request to modify microcode, and wherein the first control logic is further configured to grant or deny the request to modify microcode based on the selected entries of the one or more microcode loader enable bits. (Figure 5, steps 545-570 show an example of

granting access to update microcode and it is described in detail in column 13, lines 7 to column 14, lines 25. If the W/L bits are set to lock write access, then the request is denied. Figure 3 and column 6, lines 20-50.)

21. As per claim 21, Gafken teaches the processor of claim 16, wherein the second control logic is further coupled to receive a signal indicative of the one or more default values for the one or more microcode loader enable bits and to write the one or more default values for the one or more microcode loader enable bits into the first register in response to the reset of the processor. (Figure 3, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55. The Reset Detector handles writing default values into the first register upon a reset of the processor. The Reset Detector receives a signal indicative of a reset, which therefore is also indicative of the default values for the first register, since the default values are loaded after the indication).
22. As per claim 22, Gafken teaches the processor of claim 16, wherein the fourth control logic is coupled to receive a RESET signal in response to the reset of the processor. (Figure 3, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55.)
23. As per claim 23, Gafken teaches the processor of claim 16, further comprising: a second register (L/D) coupled to the first control logic, wherein the second register is configured to store one or more microcode loader enable lock bits. (Column 6, lines

51-59; when in lock-down mode, the microcode loader enable bit (R/L) associated with the lock-down bit is not modifiable.)

24. As per claim 24, Gafken teaches the processor of claim 23,

-Wherein the first control logic (Interface 139) is further configured to receive a request to modify microcode loader lock status: (Figure 3, figure 5, step 545, column 5, line 60 to column 6, line 3, column 6, lines 20-59, and column 13, lines 40-51)

-Wherein the first control logic is further configured to read selected entries in the one or more microcode loader enable lock bits stored in the second register in response to the request to modify microcode loader lock status: (Column 6, lines 20-59 and column 7, lines 47-54)

-And wherein the first control logic is further configured to grant or deny the request to modify microcode loader lock status based on the selected entries in the one or more microcode loader enable lock bits: (Column 6, lines 20-59 and column 7, lines 47-54)

25. As per claim 27, Gafken teaches a method for modifying microcode, the method comprising: receiving a request to modify microcode; determining microcode loader enable status (stored in W/L register); modifying microcode if the microcode loader enable status is set to enabled. (Column 6, lines 20-59, column 12, lines 29-37, figures 3 and 5, column 1, lines 18-32)

26. Given the similarities between claim 27 and claim 45, the arguments as stated for the rejection of claim 27 also apply to claim 45.

27. As per claim 28, Gafken teaches the method of claim 27, wherein determining microcode loader enable status comprises reading one or more entries corresponding to one or more microcode loader enable bits from a register. (Column 6, lines 20-59, column 12, lines 29-37, figures 3 and 5, column 1, lines 18-32; The W/L register stores one or more entries which are read in order to determine the microcode loader enable status)

28. Given the similarities between claim 28 and claim 46, the arguments as stated for the rejection of claim 28 also apply to claim 46

29. As per claim 32, Gafken teaches a method of changing microcode loader enable status, the method comprising: receiving a request to change microcode loader enable status; determining microcode loader enable lock status; and modifying microcode loader enable status if the microcode loader enable lock status is set to unlocked. (If the W/L register is not in lock-down mode, (i.e., unlocked), the loader enable status can be changed. (Column 6, lines 20-59, column 12, lines 29-37, figures 3 and 5)

30. Given the similarities between claim 32 and claim 50, the arguments as stated for the rejection of claim 32 also apply to claim 50.

31. As per claim 33, Gafken teaches the method of claim 32, wherein determining microcode loader enable lock status comprises reading one or more entries corresponding to one or more microcode loader enable lock bits from a register (L/D register). (If the W/L register is not in lock-down mode, (i.e., unlocked), the loader

enable status can be changed. (Column 6, lines 20-59, column 12, lines 29-37, figures 3 and 5).

32. Given the similarities between claim 33 and claim 51, the arguments as stated for the rejection of claim 33 also apply to claim 51.

33. As per claim 34, Gafken teaches the method of claim 32, wherein modifying microcode loader enable status comprises writing one or more entries corresponding to one or more microcode loader enable bits to a register. (The W/L register contains an entry for bits that indicate the microcode loader enable status. Writing to the W/L register can modify it. (Column 6, lines 20-59, column 12, lines 29-37, figures 3 and 5).

34. Given the similarities between claim 34 and claim 52, the arguments as stated for the rejection of claim 34 also apply to claim 52.

New Claim Rejections - 35 USC § 102

35. Claims 25-26, 29-30, 36, 43-44 and 47-48 are rejected under 35 U.S.C. 102(b) as being anticipated by Yishay et al., U.S. patent 5,704,039, herein referred to as Yishay.

36. As per claim 25, Yishay teaches a method for determining an HDT mode enable status, the method comprising:

-Receiving a request to initiate the HDT mode: (Figure 4, step 230, a write instruction causing the write bus cycle)

-Determining HDT mode enable status: (Figure 4, Step 280, is $N > M$? The HDT mode enable status is stored in the variable N)

-Initiating the HDT mode if the HDT mode enable status is set to enabled: (Figure 4, step 290, "Negate Secure" if and only if $N > M$, i.e., HDT mode enable status is set to enabled.)

37. Given the similarities between claim 25 and claim 43, the arguments as stated for the rejection of claim 25 also apply to claim 43.

38. As per claim 26, Yishay teaches the method of claim 25, wherein determining HDT mode enable status comprises reading one or more entries corresponding to one or more HDT enable bits from a register: (N is read from control circuit 46 and supplied to the selector 48 in step 220 of figure 4, and it is again read for a comparison ($N > M$?), which is the step of determining the mode enable status. Register is defined as, "A device capable of retaining information, often that contained in a small subset (for example, one word), of the aggregate information in a digital computer." (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) The variable N is retaining information over multiple iterations of the algorithm shown in figure 4, and is stored within hardware logic, therefore, reading the variable N is reading a register.

39. Given the similarities between claim 26 and claim 44, the arguments as stated for the rejection of claim 26 also apply to claim 44.

40. As per claim 29, Yishay teaches a method of changing HDT mode status, the method comprising:

-Receiving a request to change HDT mode status: (Figure 4, step 230, a write instruction causing the write bus cycle)

-Determining HDT mode enable lock status: (Figure 4, Step 280, is $N > M$? The HDT mode enable status is stored in the variable N)

-Modifying HDT mode if the HDT mode enable lock status is set to unlocked: (Figure 4, step 290, the HDT mode is changed via “Negate Secure” if $N > M$, i.e., HDT mode enable status is unlocked.))

41. Given the similarities between claim 29 and claim 47, the arguments as stated for the rejection of claim 29 also apply to claim 47.
42. As per claim 30, Yishay teaches the method of claim 29, where determining HDT mode enable lock status comprises reading one or more entries corresponding to one or more HDT enable lock bits from a register: (N is read from control circuit 46 and supplied to the selector 48 in step 220 of figure 4, and it is again read for a comparison ($N > M$?), which is the step of determining the mode enable status. Register is defined as, “A device capable of retaining information, often that contained in a small subset (for example, one word), of the aggregate information in a digital computer.” (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) The variable N is retaining information over multiple iterations of the algorithm shown in figure 4, and is inherently temporarily stored within hardware logic, therefore, reading the variable N is reading a register.)
43. Given the similarities between claim 30 and claim 48, the arguments as stated for the rejection of claim 30 also apply to claim 48.

44. As per claim 36, Yishay teaches a processor comprising:

-Means for receiving a request to initiate the HDT mode: ((Figure 4, step 230, a write instruction causing the write bus cycle)

-Means for determining HDT mode enable status: (Figure 4, Step 280, is N > M? The HDT mode enable status is stored in the variable N)

-Means for initiating the HDT mode if the HDT mode enable status is set to enabled: ((Figure 4, step 290, the HDT mode is changed via “Negate Secure” if and only if N > M, i.e., HDT mode enable status is unlocked.))

Claim Rejections - 35 USC § 103

45. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

46. Claims 3-4, 18-20, 35, 41, 42 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gafken et al., U.S. Patent 5,826,007, herein referred to as Gafken, in view of Short, Embedded Microprocessor Systems Design.

47. As per claim 3, Gafken teaches the processor of claim 1, wherein the R/L bits (HDT enable bits) are loaded upon reset after power is removed from the processor with default values, however, does not specify how the default values are written into the R/L registers by the Reset Detector 135.

48. Short teaches wherein non-volatile memory is ideal for storing data that is needed upon powering on (reset), since it will not lose its data when power is turned off. (Page 35)

49. It would have been obvious to one of ordinary skill in the art to have the default values for the R/L (HTD enable) bits stored in non-volatile memory since Short teaches that non-volatile memory is well-suited for this purpose because it retains its data even when power is shut off.

50. As per claim 4, Gafken, in view of Short, teaches the processor of claim 3, wherein the second control logic is further coupled to read the one or more default values for the one or more HDT enable bits from the one or more non-volatile memory cells and to write the one or more default values for the one or more HDT enable bits into the first register in response to the reset of the processor. (Gafken, in view of Short, teaches wherein on a reset, default values are written into the R/L register (first register), and wherein they come from non-volatile memory locations. (Gafken, column 5, lines 47-59, column 11, lines 38-55 and Short, page 35)

51. As per claim 18, Gafken teaches the processor of claim 16, wherein the W/L bits (microcode loader enable bits) are loaded upon reset after power is removed from the processor with default values, however, does not specify how the default values are written into the W/L registers by the Reset Detector 135. (Figure 3, Reset Detector, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55). Therefore, Gafken fails to teach one or more non-volatile memory cells configured to

store the one or more default values for the one or more microcode loader enable bits.

52. Short teaches wherein non-volatile memory is ideal for storing data that is needed upon powering on (reset), since it will not lose its data when power is turned off. (Page 35)
53. It would have been obvious to one of ordinary skill in the art to have the default values for the W/L (microcode loader enable) bits stored in non-volatile memory since Short teaches that non-volatile memory is well-suited for this purpose because it retains its data even when power is shut off.
54. As per claim 19, Gafken, in view of Short, teaches the processor of claim 18, wherein selected ones of the one or more non-volatile memory cells are configured to store the one or more default value for the one or more microcode loader enable bits. (Gafken, Figure 3, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55. The W/L bits (microcode loader enable bits) are loaded upon reset after power is removed from the processor with default values, by the Reset Detector 135, and in view of Short as stated above in regards to claim 18, the default values come from non-volatile memory locations.)
55. As per claim 20, Gafken teaches the processor of claim 18, wherein the second control logic is further coupled to read the one or more default values for the one or more microcode loader enable bits from the one or more non-volatile memory cells and to write the one or more default values for the one or more microcode loader

enable bits into the microcode loader register in response to the reset of the processor. (Gafken, Figure 3, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55. The W/L bits (microcode loader enable bits) are loaded upon reset after power is removed from the processor with default values, by the Reset Detector 135, and in view of Short as stated above in regards to claim 18, the default values come from non-volatile memory locations and the controlling of the transfer is done by the Reset Detector 135, which would include a read and write (read to get the value out of the non-volatile memory cell and write to get it into a W/L register.)

56. As per claim 35, Gafken teaches a method of operating a processor, the method comprising:

-Writing one or more default values as one or more various entries in one or more registers in response to a reset of the processor, wherein the one or more various entries are selected from the group consisting of: one or more HDT enable bits, one or more HDT enable lock bits, one or more microcode loader enable bits, and one or more microcode loader enable lock bits. (The Reset Detector stores default values in the R/L register (HDT enable status), W/L register (microcode loader enable status), and L/D register (HDT enable lock and microcode loader enable lock). The appropriate default values are selected and put in the appropriate registers upon a reset. Figure 7, Column 11, lines 3-55, column 5, lines 47-59)

57. However, Gafken does not specify how the default values are written into the W/L registers by the Reset Detector 135. (Figure 3, Reset Detector, Column 5,

lines 47-59, and column 10, line 62 to column 11, lines 3-55). Therefore, Gafken fails to teach obtaining one or more default values, wherein obtaining the one or more default values is selected from the group consisting of: reading the one or more default values from one or more non-volatile memory cells, and receiving the one or more default values as a strapped value through a pull-up or pull-down resistor;

58. Short teaches wherein non-volatile memory is ideal for storing data that is needed upon powering on (reset), since it will not lose its data when power is turned off. (Page 35)

59. It would have been obvious to one of ordinary skill in the art to have the default values for the R/L (HDT enable) bits, W/L (microcode loader enable) bits and L/D (HDT and microcode loader lock enable) bits stored in non-volatile memory (and therefore read out/obtained and placed in the appropriate registers) since Short teaches that non-volatile memory is well-suited for this purpose because it retains its data even when power is shut off.

60. Given the similarities between claim 35 and claim 53, the arguments as stated for the rejection of claim 35 also apply to claim 53.

61. As per claim 41, Gafken teaches a processor, comprising:
-Means for storing one or more default values, wherein the default values are selected from the group consisting of: HDT enable status, HDT enable lock status, microcode loader enable status, and microcode loader enable lock status: (The Reset

Detector stores default values in the R/L register (HDT enable status), W/L register (microcode loader enable status), and L/D register (HDT enable lock and microcode loader enable lock). The appropriate default values are selected and put in the appropriate registers upon a reset. Figure 7, Column 11, lines 3-55, column 5, lines 47-59)

-And means for writing the one or more default values as one or more various entries in the means for storing the one or more default values in response to a reset of the processor, wherein the one or more various entries are selected from the group consisting of: one or more HDT enable bits, one or more HDT enable lock bits, one or more microcode loader enable bits, and one or more microcode loader enable lock bits: (The Reset Detector stores default values in the R/L register (HDT enable status), W/L register (microcode loader enable status), and L/D register (HDT enable lock and microcode loader enable lock). The appropriate default values are selected and put in the appropriate registers upon a reset. Figure 7, Column 11, lines 3-55, column 5, lines 47-59)

62. However, Gafken does not specify how the default values are written into the W/L registers by the Reset Detector 135. (Figure 3, Reset Detector, Column 5, lines 47-59, and column 10, line 62 to column 11, lines 3-55). Therefore, Gafken fails to teach means for obtaining the one or more default values, wherein obtaining the one or more default values is selected from the group consisting of:

-Reading the one or more default values from non-volatile memory,

-And receiving the one or more default values as a strapped value through a pull-up or pull-down resistor;

63. Short teaches wherein non-volatile memory is ideal for storing data that is needed upon powering on (reset), since it will not lose its data when power is turned off. (Page 35)
64. It would have been obvious to one of ordinary skill in the art to have the default values for the R/L (HDT enable) bits, W/L (microcode loader enable) bits and L/D (HDT and microcode loader lock enable) bits stored in non-volatile memory since Short teaches that non-volatile memory is well-suited for this purpose because it retains its data even when power is shut off.
65. As per claim 42, Gafken teaches a computer system, comprising: a processor, comprising:

-Means for storing one or more default values, wherein the default values are selected from the group consisting of: HDT enable status, HDT enable lock status, microcode loader enable status, and microcode loader enable lock status: (The Reset Detector stores default values in the R/L register (HDT enable status), W/L register (microcode loader enable status), and L/D register (HDT enable lock and microcode loader enable lock). The appropriate default values are selected and put in the appropriate registers upon a reset. Figure 7, Column 11, lines 3-55, column 5, lines 47-59)

-And means for writing the one or more default values as one or more various entries in the means for storing the one or more default values in response to a reset of the processor, wherein the one or more various entries are selected from the group consisting of: one or more HDT enable bits, one or more HDT enable lock bits, one or more microcode loader enable bits, and one or more microcode loader enable lock bits: (The Reset Detector stores default values in the R/L register (HDT enable status), W/L register (microcode loader enable status), and L/D register (HDT enable lock and microcode loader enable lock). The appropriate default values are selected and put in the appropriate registers upon a reset. Figure 7, Column 11, lines 3-55, column 5, lines 47-59)

-A bridge coupled to the processor: (Figures 1 and 2 show the system of Gafken has multiple hardware devices connected through multiple buses. A bridge is defined as, "A hardware adapter that forwards transactions between buses." (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) Data transactions do occur between multiple devices on the bus, and there is inherently hardware to do so, therefore there is inherently a bridge present to forward transactions between buses. (Column 3, lines 38-65)

-And a memory operable coupled to the processor, wherein the memory is configured to store BIOS code: (Figure 5, column 12, lines 19 to column 14, line 26)

66. However, Gafken does not specify how the default values are written into the W/L registers by the Reset Detector 135. (Figure 3, Reset Detector, Column 5,

lines 47-59, and column 10, line 62 to column 11, lines 3-55). Therefore, Gafken fails to teach means for obtaining the one or more default values, wherein obtaining the one or more default values is selected from the group consisting of:

-Reading the one or more default values from non-volatile memory,

-And receiving the one or more default values as a strapped value through a pull-up or pull-down resistor;

67. Short teaches wherein non-volatile memory is ideal for storing data that is needed upon powering on (reset), since it will not lose its data when power is turned off. (Page 35)

68. It would have been obvious to one of ordinary skill in the art to have the default values for the R/L (HDT enable) bits, W/L (microcode loader enable) bits and L/D (HDT and microcode loader lock enable) bits stored in non-volatile memory (and therefore read out and placed in the appropriate registers) since Short teaches that non-volatile memory is well-suited for this purpose because it retains its data even when power is shut off.

69. Claims 31, 37-40 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yishay et al., U.S. Patent 5,826,007, herein referred to as Yishay, in view of Kime, Logic and Computer Design Fundamentals, 2nd ed.

70. As per claim 31, Yishay teaches the method of claim 29, wherein modifying HDT mode status comprises writing one or more entries corresponding to one or more HDT enable bits to a register: (When the "Secure Signal" is negated, it is held over

many cycles (the signal is negated throughout a debugging process, including external reads of internal memory and registers, which inherently would be over many clock cycles, column 9, lines 44-54)).

71. However, there is no explicit teaching of the 'Secure Signal' value that is output from the control circuit being stored in a register whether in the control circuit or in each of the individual pieces or hardware that receive the signal.
72. Kime teaches wherein flip-flops store values and connect different pieces of hardware logic together to ensure synchronization despite differences in circuit delay and simple logic design. (Pages 185, 186.) Register is defined as, "A device capable of retaining information, often that contained in a small subset (for example, one word), of the aggregate information in a digital computer." (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) A flip-flop is a register.)
73. It would have been obvious to one of ordinary skill in the art to use the flip-flop, as taught in Kime, to act as a buffer and synchronizer between the control circuit's output, "Secure", and the other pieces or hardware on the chip. This would cause proper operation of the circuits with respect to the "Secure" signal's value despite any differences in circuit delay that may exist, and it is a "relatively easy" solution.
74. Given the similarities between claim 31 and claim 49, the arguments as stated for the rejection of claim 31 also apply to claim 49.
75. As per claim 37, Yishay teaches the processor of claim 36, further comprising:
 - Means for storing an indication of the HDT mode status: (When the "Secure Signal" is negated, it is held over many cycles (the signal is negated throughout a

debugging process, including external reads of internal memory and registers, which inherently would be over many clock cycles, column 9, lines 44-54)).

76. However, there is no explicit teaching of the 'Secure Signal' value that is output from the control circuit being stored in a register whether in the control circuit or in each of the individual pieces or hardware that receive the signal.

77. Kime teaches wherein flip-flops store values and connect different pieces of hardware logic together to ensure synchronization despite differences in circuit delay and simple logic design. (Pages 185, 186.) Register is defined as, "A device capable of retaining information, often that contained in a small subset (for example, one word), of the aggregate information in a digital computer." (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) A flip-flop is a register.)

78. It would have been obvious to one of ordinary skill in the art to use the flip-flop, as taught in Kime, to act as a buffer and synchronizer between the control circuit's output, "Secure", and the other pieces or hardware on the chip. This would cause proper operation of the circuits with respect to the "Secure" signal's value despite any differences in circuit delay that may exist, and it is a "relatively easy" solution.

79. As per claim 38, Yishay, in view of Kime, teaches the processor of claim 37, further comprising:

-Means for providing the means for storing with one or more default values for the indication of the HDT mode status: (The "Secure Mask 66" provides default values for the Secure signal, column 9, lines 44 to column 10, line 11 and Figure 3)

80. As per claim 39, Yishay, in view of Kime, teaches the processor of claim 38, further comprising:

-Means for receiving a request to change HDT mode status: (Figures 3 and 4, step 230, a write instruction causing the write bus cycle causes data and address signals on the data and address signal lines 54 and 56)

-Means for determining HDT mode lock status: (Figure 4, Step 280, is $N > M$?)

The HDT mode enable status is stored in the variable N, if $N > M$, the HDT mode lock status becomes unlocked)

-Means for modifying HDT mode status if the HDT mode lock status is set to unlocked: (Figure 4, step 290, the HDT mode is changed via "Negate Secure" if $N > M$, i.e., HDT mode enable status is unlocked.)

81. As per claim 40, Yishay, in view of Kime, teaches the processor of claim 39, further comprising: means for storing an indication of the HDT mode lock status: (N is read from control circuit 46 and supplied to the selector 48 in step 220 of figure 4, and it is again read for a comparison ($N > M$?), which is the step of determining the mode enable status. Register is defined as, "A device capable of retaining information, often that contained in a small subset (for example, one word), of the aggregate information in a digital computer." (The Authoritative Dictionary of IEEE Standards Terms, 7th ed.) The variable N is retaining information over multiple iterations of the algorithm shown in figure 4, and is inherently temporarily stored within hardware logic, therefore, reading the variable N is reading a register.)

Response to Arguments

82. Applicants arguments filed on 7/26/2005 have been fully considered but they are not persuasive.

83. Applicant argues the novelty/rejection of claims 1, 35, 41-42 and 53.

“The Examiner the alleges that a read operation is equivalent to a HDT operation. Applicant respectfully disagrees. HDT operations are well known in the art and persons of ordinary skill in the art will appreciate that simply preventing or allowing a memory read operation directed to a particular block of memory is not the same as enabling an HDT operation. Thus, Applicant submits that Gafken fails to teach or suggest HDT enable bits or HDT enable lock bits.”

84. These arguments are not found persuasive for the following reasons:

a) To clarify, applicant’s attention is directed towards the specification. No explicit definition of a Hardware-Debug-Test (HDT) has been provided. Applicant has asserted that HDT has a well-known meaning in the art that differentiates it from a read-from-memory operation, however, Applicant has not provided supporting evidence of this. Examiner notes that searches in “The Authoritative Dictionary of IEEE Standards Terms, 7th Ed.”, online dictionary at www.dictionary.com and “Acronym Finder” on www.dictionary.com all fail to provide a definition for “Hardware-Debug-Test” or “HDT”. (See attached references). Therefore, in light of the specification’s lack of a definition, lack of evidence cited by Applicant, and the evidence found by Examiner, enable bits for a read operation of memory satisfies the claim languages, “HDT enable bits” in its broadest interpretation. See also the further clarified 35 U.S.C. 102 Rejections above.

85. Applicant argues the novelty/rejection of claims 16, 27, 32, 45 and 50.

"Gafken describes a lock bit array 315 that includes a write lock bit, which indicates whether a corresponding block of memory is locked to prevent write or erase operations, or unlocked. A block of memory array 130 that is write locked is prevented from being accessed for program or erase operations. See Gafken, col. 6, lines 19-40. However, Gafken is completely silent with regard to enabling a microcode loader, e.g., a unit or devices that may load microcode. Accordingly, Applicant respectfully submit[s] that the invention set forth in claims 16, 27, 32, 45 and 50, and all claims depending therefrom is not anticipated by Gafken."

86. These arguments are not found persuasive for the following reasons:
 - a) The memory locations that are locked in some instances store microcode. See Col. 1, line 13 to col. 2, line 9. The locks are in place to prevent or allow *updating* or erasing of microcode stored in the memory locations. Therefore, when microcode is allowed to be updated, there is an inherent enabling of a microcode loader. (See also figure 5, "Update Code" is mentioned in many steps of the flow diagram.) There must be circuitry to implement the loading of microcode, for instance, the BIOS Update Utility 151 on Mass Storage 118 is a "microcode loader", which would at least anticipate the independent claims. However, in the present interpretation of Gafken for all the claims currently being argued, the Interface 139 is the "microcode loader." (Figure 1, figure 3 and col. 5, line 60 to col. 6, line 3)).
87. Applicant argues the 35 U.S.C. 103 rejections to claims 3-4, 18-20, 35, 41-42 and 53. However, since the above arguments in regard to the 35 U.S.C. 102 Rejections under Gafken were found not to be persuasive, the arguments for claims 3-4, 18-20, 35, 41-42 and 53, which relied solely on the above arguments related to microcode loader and HDT enable bits, are also found not to be persuasive.

88. Applicant argues the novelty/rejection of claims. 25-26, 29-30, 36, 43-44 and 47-48. However, as stated in ¶ 3, the rejections under Sakaki have been withdrawn and the corrected 35 U.S.C. 102 and 103 Rejections under Yishay have been established.

Conclusion

89. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

90. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin P Rizzuto whose telephone number is (571) 272-4174. The examiner can normally be reached on M-F, 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KPR



EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100